

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(51) Int.Cl. <sup>8</sup>	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 9/32	3 1 0 J			
13/28	3 1 0 A	9172-5E		
15/78	5 1 0 C			

審査請求 未請求 請求項の数1 FD (全 12 頁)

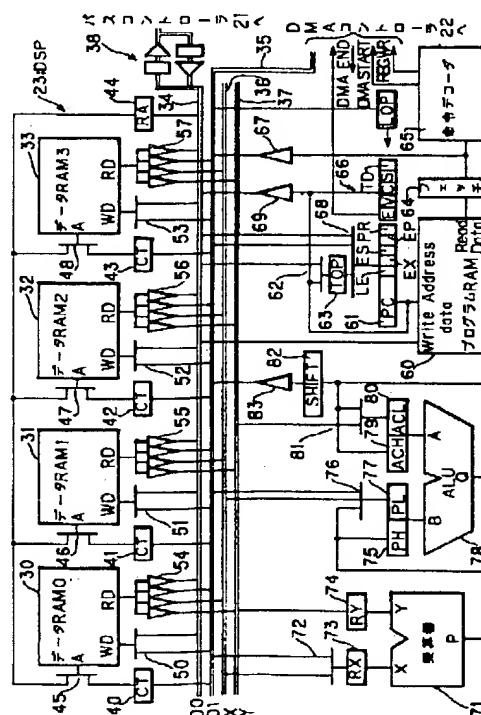
(21)出願番号	特願平6-274426	(71)出願人	000004075 ヤマハ株式会社 静岡県浜松市中沢町10番1号
(22)出願日	平成6年(1994)10月13日	(72)発明者	森本 実 静岡県浜松市中沢町10番1号 ヤマハ株式 会社内
		(72)発明者	吉田 佳司 静岡県浜松市中沢町10番1号 ヤマハ株式 会社内
		(74)代理人	弁理士 伊丹 勝

(54) 【発明の名称】 デジタル信号処理装置

(57) 【要約】

【目的】 小容量のプログラムメモリでも効率良く長いプログラムを実行させることができ、CPUの負担も軽減させる。

【構成】 プログラム RAM 60 に格納されるプログラムに、演算命令と共にプログラムの DMA 命令を含ませておく。命令デコーダ 65 は、プログラム RAM 60 からプログラムの各命令を順次読み出して各部を制御すると共に、このプログラムの実行過程でプログラムの DMA 命令が解読された場合に、プログラム・カウンタ 61 の値を退避させたのち、外部からプログラム RAM 60 にプログラムを DMA 転送させるため DMA コントローラ 22 を起動する。そして、転送終了後にプログラム・カウンタ 61 の値を復帰させてプログラムの実行を再開させる。これにより、DSP 23 は、自らのプログラムで自らのプログラムを補充していくように動作する。



1

## 【特許請求の範囲】

【請求項 1】 プログラムを格納するプログラムメモリと、  
このプログラムメモリからプログラムの各命令を順次読み出して解読する命令解読手段と、  
この命令解読手段によって制御されてデータの演算処理を実行する演算処理手段とを備えたデジタル信号処理装置において、  
前記プログラムメモリに格納されるプログラムに、演算命令と共にプログラムの DMA 命令を含ませておき、  
前記命令解読手段は、前記プログラムメモリからプログラムの各命令を順次読み出して各部を制御すると共に、  
このプログラムの実行過程で前記プログラムの DMA 命令が解読された場合に、プログラム・カウンタの値を退避させたのち外部から前記プログラムメモリにプログラムをダイレクト・メモリ・アクセスにより転送させる制御手段を起動して、転送終了後に前記プログラム・カウンタの値を復帰させて前記プログラムの実行を再開させることを特徴とするデジタル信号処理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 この発明は、デジタル・シグナル・プロセッサ（DSP）等の演算動作を規定するプログラムを外部装置から内部のプログラムメモリにダイレクト・メモリ・アクセス（DMA）転送するデジタル信号処理装置に関する。

## 【0002】

【従来の技術】 画像処理装置や音源装置等で使用される DSP は、内部に備えられたデータ RAM（ランダム・アクセス・メモリ）に外部から供給される係数データ及び信号データを一旦格納し、内部の乗算器及び ALU（算術論理ユニット）でこれらのデータの演算処理を実行したのち、その演算結果をデータ RAM に格納し外部に出力することにより、フィルタリング等の各種処理を実行する。これらの演算処理を規定する演算プログラムは、実行前に CPU から DSP 内部のプログラム RAM に前もって格納される。

## 【0003】

【発明が解決しようとする課題】 この種の DSP では、基本的にはプログラム実行中のプログラム RAM に対するアクセスは不可能であるため、長いプログラムを実行する場合、十分な容量のプログラム RAM を DSP の内部に持つか、一旦、DSP の実行を強制的に停止して、CPU から新たなプログラムを書き込む必要がある。しかしながら、前者はハードウェアのコスト上昇を招き、後者は非効率的で CPU の負担が増す。

【0004】 この発明は、このような問題点に鑑みなされたもので、小容量のプログラムメモリでも効率良く長いプログラムを実行させることができ、CPU に負担をかけることもないデジタル信号処理装置を提供するこ

2

とを目的とする。

## 【0005】

【課題を解決するための手段】 この発明に係るデジタル信号処理装置は、プログラムを格納するプログラムメモリと、このプログラムメモリからプログラムの各命令を順次読み出して解読する命令解読手段と、この命令解読手段によって制御されてデータの演算処理を実行する演算処理手段とを備えたデジタル信号処理装置において、前記プログラムメモリに格納されるプログラムに、演算命令と共にプログラムの DMA 命令を含ませておき、前記命令解読手段は、前記プログラムメモリからプログラムの各命令を順次読み出して各部を制御すると共に、このプログラムの実行過程で前記プログラムの DMA 命令が解読された場合に、プログラム・カウンタの値を退避させたのち外部から前記プログラムメモリにプログラムをダイレクト・メモリ・アクセスにより転送させる制御手段を起動して、転送終了後に前記プログラム・カウンタの値を復帰させて前記プログラムの実行を再開させることを特徴とする。

## 【0006】

【作用】 この発明に係るデジタル信号処理装置によれば、内部のプログラムに、演算命令と共にプログラムの DMA 命令を含ませておき、プログラムの実行過程で、プログラムの DMA 命令が解読された場合に、プログラム・カウンタ値を退避させたのち、外部の DMA 制御手段等を起動して内部のプログラムメモリに新たなプログラムを転送補充して、プログラム・カウンタ値を復帰させたのち、プログラムを再開させるようにしているので、デジタル信号処理装置が自らのプログラムで自らのプログラムを補充していくように動作する。このため、ホスト CPU が新たなプログラムの書き込みに全く関与する必要がなく、CPU の負担は大幅に軽減されると共に、容量の小さなプログラムメモリであっても、効率良く長いプログラムの実行が可能になる。また、この発明によれば、CPU が介入する初期セット時においても、必要最小限度のプログラムだけを転送しておけば良く、これによる CPU の負担軽減も図れる。

## 【0007】

【実施例】 以下、図面を参照して、この発明の実施例について説明する。図 1 は、この発明の一実施例に係るゲーム機器のシステム構成を示すブロック図である。このシステムは、システム・コントロール・ユニット（以下、SCU と呼ぶ）1 によって調停される 3 つのバス 2, 3, 4 に、各種の機能要素を接続して構成されている。バス 2 には、システム全体の制御を司る CPU 5 と、この CPU 5 の作業領域を提供するワーク RAM 6 と、システム起動時の処理プログラムを記憶したブート ROM（リード・オンリー・メモリ）7 とが接続されている。バス 3 には、ゲーム ROM 9 等のゲームソースが接続されている。バス 4 には、画像プロセッサ 11 と音

3

源プロセッサ 12 とが接続されている。

【0008】SCU 1 には、バス 2, 3, 4 の切換え制御を実行するバスコントローラ 21 と、内部のメモリと外部の回路との間でデータ、プログラム等を DMA 転送する DMA コントローラ 22 と、内部のメモリに格納された係数データと信号入力データとの積和演算処理を実行する DSP 23 とが設けられている。

【0009】図 2 は、DSP 23 の更に詳細な構成を示すブロック図である。この DSP 23 には、図面上では 0, 1, 2, 3 の番号が付加された 4 つのデータ RAM 30, 31, 32, 33 が備えられている。これらのデータ RAM 30~33 は、4 つの独立したバス 34, 35, 36, 37 からそれぞれ独立にアクセスが可能になっている。即ち、4 つのバス 34~37 のうち、D0 バス 34 は、双方向バッファ 38 を介してバスコントローラ 21 に接続されており、バスコントローラ 21 を介して外部のバス 2, 3, 4 との間でデータのやりとりが行えるようになっている。D1 バス 35、X バス 36、Y バス 37 は、DSP 23 の内部の演算用バスであるが、そのうち D1 バス 35 は、DMA コントローラ 22 にも接続されている。

【0010】各データ RAM 30~33 のアドレス端子には、各データ RAM 30~33 毎に設けられて D1 バス 35 に共通に接続されたアドレスカウンタ (CT) 40, 41, 42, 43 からのアドレスと、D0 バス 34 に接続された外部からの共通メモリアクセスに使用されるアドレスレジスタ (RA) 44 からのアドレスのいずれか一方が、それぞれ選択回路 45, 46, 47, 48 で選択されて与えられる。また、各データ RAM 30~33 には、D0 バス 34 上のデータと D1 バス 35 上のデータのうち、選択回路 50, 51, 52, 53 で選択されたいずれか一方のデータが格納される。また、各データ RAM 30~33 からは、4 つのバス 34~37 のうち、ゲート回路 54, 55, 56, 57 を介して選択された任意のバスにデータが読み出される。

【0011】プログラム RAM 60 は、この DSP 23 の実行を規定するプログラムを記憶するメモリで、外部の例えばワーク RAM 7 からバス 2、バスコントローラ 21、双方向バッファ 38 及び D0 バス 34 を介して DMA 転送されたプログラムを記憶する。プログラム RAM 60 のアドレスは、プログラムカウンタ (PC) 61 によって与えられる。PC 値は、初期設定時には、外部の CPU 5 から D0 バス 34 及び選択回路 68 を介して TOP レジスタ 63 に格納され、プログラムの DMA 転送時は、D1 バス 35 及び選択回路 68 を介して格納され、DMA 実行に先だって選択回路 62 を介して TOP レジスタ 63 に退避される。

【0012】PC 値に従ってプログラム RAM 60 から読み出された命令コードは、フェッチ回路 64 でフェッチされ、命令デコーダ 65 で解読される。命令デコーダ

4

65 は、この解読結果に基づいて DSP 23 の各部を制御すると共に、E, V, C, S の各種フラグ 66 を書き換える。また、命令デコーダ 65 は、DMA コントローラ 22 に対するパラメータの設定や DMA の起動制御も実行する。この場合、フェッチ回路 64 にフェッチされたパラメータは、ゲート回路 67 及び D1 バス 35 を介して DMA コントローラ 22 に転送される。また、PC 値及び各種のフラグ 66 は、選択回路 68 を介して TOP レジスタ 63 及び D0, D1 バス 34, 35 から供給されたり、書き換えられたりすると共に、ゲート回路 69 及び D0 バス 34 を介して外部に転送される。

【0013】乗算器 71 は、選択回路 72 で選択され、RX レジスタ 73 に格納された D1 バス 35 上又は X バス 36 上のデータと、RY レジスタ 74 に格納された Y バス 37 上のデータとを乗算する。その乗算結果は、上位ビットが PH レジスタ 75、下位ビットが選択回路 76 を介して PL レジスタ 77 に格納される。PL レジスタ 77 には、D1 バス 35、X バス 36 上のデータも選択回路 76 で選択されて格納されるようになっている。ALU 78 は、PH, PL レジスタ 75, 77 に格納されたデータと、ACH, ACL レジスタ 79, 80 に格納されたデータとを加算する。その加算結果は、上位ビットが ACH レジスタ 79 に、下位ビットが選択回路 81 を介して ACL レジスタ 80 に格納されると共に、シフトレジスタ 82 及びゲート回路 83 を介して D1 データに出力される。この構成により、積和演算等の演算処理を実行することができる。

【0014】次に、このシステムの動作について説明する。CPU 5 が、先ず、ワーク RAM 6 に格納されている DSP 23 の実行用のプログラムを DSP 23 のプログラム RAM 60 に転送する。このとき、DSP 23 のプログラム RAM 60 の容量には制限があるので、プログラム RAM 60 が記憶可能な量だけプログラムが転送される。次に、CPU 5 は、DSP 23 のプログラムを起動する PC 値を PC 61 に転送する。更に、実行フラグ EX を 1 にして、プログラムを開始させる。

【0015】DSP 23 のプログラムが起動されたら、以後は CPU 5 の制御によらず、DSP 23 が独立して以下の処理を実行する。図 3 は、プログラム RAM 60 に格納されたプログラムの一例を示す図、図 4 はそのプログラムにより実行される処理の内容を示すフローチャートである。

【0016】まず、起動直後においては、データ RAM 30~33 に係数パラメータ、演算用信号データ等の DSP 処理に必要なデータが全く格納されていない状態であるため、これらのデータをデータ RAM 30~33 に格納する必要がある。このため、プログラムの先頭番地 "0" に、DMAC パラメータ設定命令を配置し、続いて DMA 命令、DMA 終了確認命令を配置する。命令デコーダ 65 は、DMAC パラメータ設定命令を解読する

5

と、続くDMACパラメータをD1バス35に出力すると共に、DMAコントローラ22にレジスタライト信号を出力する。DMACパラメータは、例えば、転送元アドレス、転送先アドレス及び転送ワード数からなり、転送元アドレスとして例えばワークRAM6の係数データの格納領域の先頭番地、転送ワード数として係数データ数がDMAコントローラ22に設定され、転送先アドレスとしてデータRAM30の先頭番地がアドレスカウンタ40に設定される。そして、命令デコーダ65がDMA命令を解釈すると、DMAコントローラ22にDMAスタート信号が出力される。これにより、DMAコントローラ22は、外部のワークRAM6から内部のデータRAM30へ係数データのDMA転送を実行する(S11)。

【0017】DMAが開始されると、TOフラグは“1”になり、以後、DMAコントローラ22からDMA終了信号が出力されるまで、TOフラグは“1”を維持する。続く、DMA終了確認命令では、このTOフラグが“0”になるまで、待ち状態となる(S2)。同様の処理により、データRAM31に外部から演算入力データをDMA転送する(S3、S4)。

【0018】更に、同様の処理により、データRAM32に外部から演算入力データをDMA転送するが(S5)、この状態では、既にデータRAM30、31に係数データ及び演算入力データがそれぞれ格納されているので、これらのデータ間の演算処理は実行可能である。そこで、データRAM32に対するDMA転送が開始されたら、直ちにデータRAM30に格納された係数データと、データRAM31に格納された演算入力データとを、乗算器71及びALU78を使用して積和演算処理し、その演算結果をデータRAM33に格納する処理を実行する(S6)。この場合、図5(a)に示すように、バス上でデータが衝突しないように、データRAM32へのDMA転送はD0バス34を介して行い、データRAM30からの係数データの読み出しはXバス36、データRAM31からの演算入力データの読み出しはYバス37をそれぞれ介して行い、データRAM33への演算結果の書き込みはD1バス35を介して行うというように、各データの転送経路をそれぞれ独立させる。これにより、データRAM32へのDMA転送とデータRAM30、31、33による演算処理とを並列に実行することができる。

【0019】一連の演算処理が終了した時点で、DMA終了確認命令が実行される(S7)。図3に示すように、DMA終了確認時T1では、DMA処理が終了(T2で終了)しているケースが殆どであり、次の命令の待ち時間は実質的に無くなる。次に、データRAM33に格納された演算結果を、D0バス34を介して外部にDMA転送する(S8)。この様子を図5(b)に示す。DMA終了確認がなされたら(S9)、データRAM3

6

1への演算入力データのDMA転送を開始し(S10)、続いてデータRAM30の係数データとデータRAM32の演算入力データとの演算処理を実行し、その演算結果をデータRAM33に格納する(S11)。この様子を図5(c)に示す。演算処理終了後、DMA終了の確認がなされたら(S12)、データRAM33に格納された演算結果を外部にDMA転送する(S13)。DMA終了確認がなされたら(S14)、以後、ステップS5～S14を繰り返す。

【0020】また、例えば、図6に示すように、データRAM33から外部へのDMA転送時に、データRAM30に格納された係数データとデータRAM32に格納された演算入力データとを演算処理し、その演算結果をデータRAM31に格納する処理を並行させる。そして、データRAM33のDMA転送が終了する時点T3を見計らって、DMA終了確認命令を配置させ、続いて外部からデータRAM33に新たな演算入力データをDMA転送するようにしてもよい。このときの、データの流れを図7に示す。演算処理の時間に比べてDMA転送時間が十分に短い場合、図7(b)、(c)でそれぞれ示すデータRAM33に対する2回のDMA転送と演算処理とを並行させることにより、更に効率の良い処理が可能である。この場合、次の演算処理は、データRAM30とデータRAM33とで行い、その間、データRAM31のデータを入れ替えることになる。

【0021】なお、以上の処理では、DMA終了を確認する命令をプログラム中に挿入したが、特にDMA転送と演算処理とを同時に実行させる場合、DMAに要する時間が予め分かっているならば、DMA開始からDMA転送に要する時間だけ経過した時点で指していると予想されるPC値よりも後に、次のDMA転送命令や演算命令を配置しておくことにより、DMA終了確認命令を省くこともできる。

【0022】ところで、以上の処理を実現するためには、各データRAM30～33にアドレスを与えるアドレスカウンタ(CT)40～43の値は、DSP23の内部の命令によって書換可能であり、この内部及びDMAコントローラ22の双方からインクリメント可能であることが必要である。このようにDSP23とDMAコントローラ22の双方からCT40～43の値を制御可能であると、DMA転送と演算処理とを並行させるというシステムの性格上、ソフトウェアの記述によっては、DMA転送中のデータRAMと、演算命令でアクセスされるデータRAMとの重複(オーバーヘッド)が発生することがある。そこで、各データRAM30～33にアドレスを与えるCT40～43のそれぞれに、そのデータRAMがDMA転送のために選択されたことを示すSEL0～SEL3フラグを設けると共に、図2に示した回路に図8に示すようなオーバーヘッド防止回路を追加することが望ましい。

【0023】このオーバーヘッド防止回路では、DMA Cパラメータから得られるDATA RAM選択信号と、命令デコーダ65からのDMA開始信号とのAND出力をゲート回路90～93で得、各ゲート回路90～93の“1”出力でフリップフロップ回路94～97をセットし、DMAコントローラ22から出力されるDMA終了信号でフリップフロップ回路94～97をリセットする。これにより、データRAM30～33のうち、DMA実行中のデータRAMに対応したフリップフロップ94～97から“1”出力が得られる。このフリップフロップ94～97の出力であるSEL0～SEL3に基づき、ゲート回路100～103で、命令デコーダ65から出力される各データRAM30～33のアクセス信号をゲートすれば、DMA実行中に命令デコーダ65からアクセス指示されたデータRAMに対応するゲート回路100～103から“1”が出力される。これをゲート回路104で検出したら、PC待機信号を出力してプログラムカウンタ61を停止させ、プログラムの実行を待機させる。これにより、DMA実行中のデータRAMに対応するアドレスカウンタ40～43の書換も含めて、そのデータRAMに対するいかなるアクセスも不可能になる。この構成によれば、オーバーヘッドを意識せずにプログラムを記述することができる。

【0024】次に、プログラムのDMA転送について説明する。前述したように、DSP23に内蔵されたプログラムRAM60に容量的な制限があると、DSP23が実行すべき全てのプログラムを一度にプログラムRAM60に格納することができない。そこで、DSP23のプログラム中に、図9に示すような、プログラムRAM60へのプログラムのDMA転送命令を記述する。このプログラムのDMA命令が実行された場合の処理を図10に示す。

【0025】プログラムのDMA命令が実行されると、まず、現在のPC値がTOPレジスタ63に退避される(S21)。即ち、このDMA命令実行時点でフェッチ回路64には、次のDMACパラメータ設定命令がプリフェッチされているので、PC値は、DMACパラメータ設定命令の次のアドレス(図9のN)を指している。TOPレジスタ63には、このNが格納されることになる。続くDMACパラメータ設定命令によって、転送元アドレス、転送先アドレス及び転送命令数がDMAコントローラ22に設定され、プログラムカウンタ61に転送先アドレスが設定される(S22)。次に、設定された命令数が外部からD0バス34を介してプログラムRAM60にDMA転送される(S23)。例えば、転送先アドレスを図9に示すように、DMACパラメータ設定命令の次の番地Nに設定した場合には、DMACパラメータ設定命令の次の番地から新たな命令A、B、…が順番に格納されていくことになる。そして、設定された数のDMA転送が終了したら、TOPレジスタ63に退

避されていたPC値をプログラムカウンタ61にセットする(S24)。以後、DMACパラメータ設定命令の次の命令Aからプログラムの実行が再開されることになる。

【0026】このように、DSP23自身のプログラムによって、自分のプログラムをDMA転送することにより、プログラムRAM60の容量に拘らず、CPU5の負担を全く増すことなく、どのような長いプログラムでも連続して実行させることができる。

#### 【0027】

【発明の効果】以上述べたように、この発明によれば、内部のプログラムに、演算命令と共にプログラムのDMA命令を含ませておき、プログラムの実行過程で、プログラムのDMA命令が解読された場合に、プログラム・カウンタ値を退避させたのち、外部のDMA制御手段等を起動して内部のプログラムメモリに新たなプログラムを転送補充して、プログラム・カウンタ値を復帰させたのち、プログラムを再開させるようにしているので、デジタル信号処理装置が自らのプログラムで自らのプログラムを補充していくように動作し、ホストCPUが新たなプログラムの書き込みに全く関与する必要がなく、CPUの負担は大幅に軽減されると共に、容量の小さなプログラムメモリであっても、効率良く長いプログラムの実行が可能になるという効果を奏する。

#### 【図面の簡単な説明】

【図1】 この発明の一実施例に係るゲーム機器のシステム構成を示すブロック図である。

【図2】 同システムにおけるDSPの詳細ブロック図である。

【図3】 同システムにおけるプログラムRAMに格納されるプログラムの一例を示す図である。

【図4】 同プログラムにより実行される処理の内容を示すフローチャートである。

【図5】 同処理におけるデータの流れを説明するための図である。

【図6】 同システムにおけるプログラムRAMに格納されるプログラムの他の例を示す図である。

【図7】 同プログラムにより実行される処理におけるデータの流れを説明するための図である。

【図8】 同システムにおけるオーバーヘッド防止回路の回路図である。

【図9】 同システムにおけるプログラムのDMA転送命令を用いたプログラムの例を示す図である。

【図10】 同プログラムによって実行される処理を示すフローチャートである。

#### 【符号の説明】

1…システム・コントロール・ユニット、2、3、4…バス、5…CPU、6…ワークRAM、7…ブートROM、9…ゲームROM、11…画像プロセッサ、12…音源プロセッサ、21…バスコントローラ、22…DM

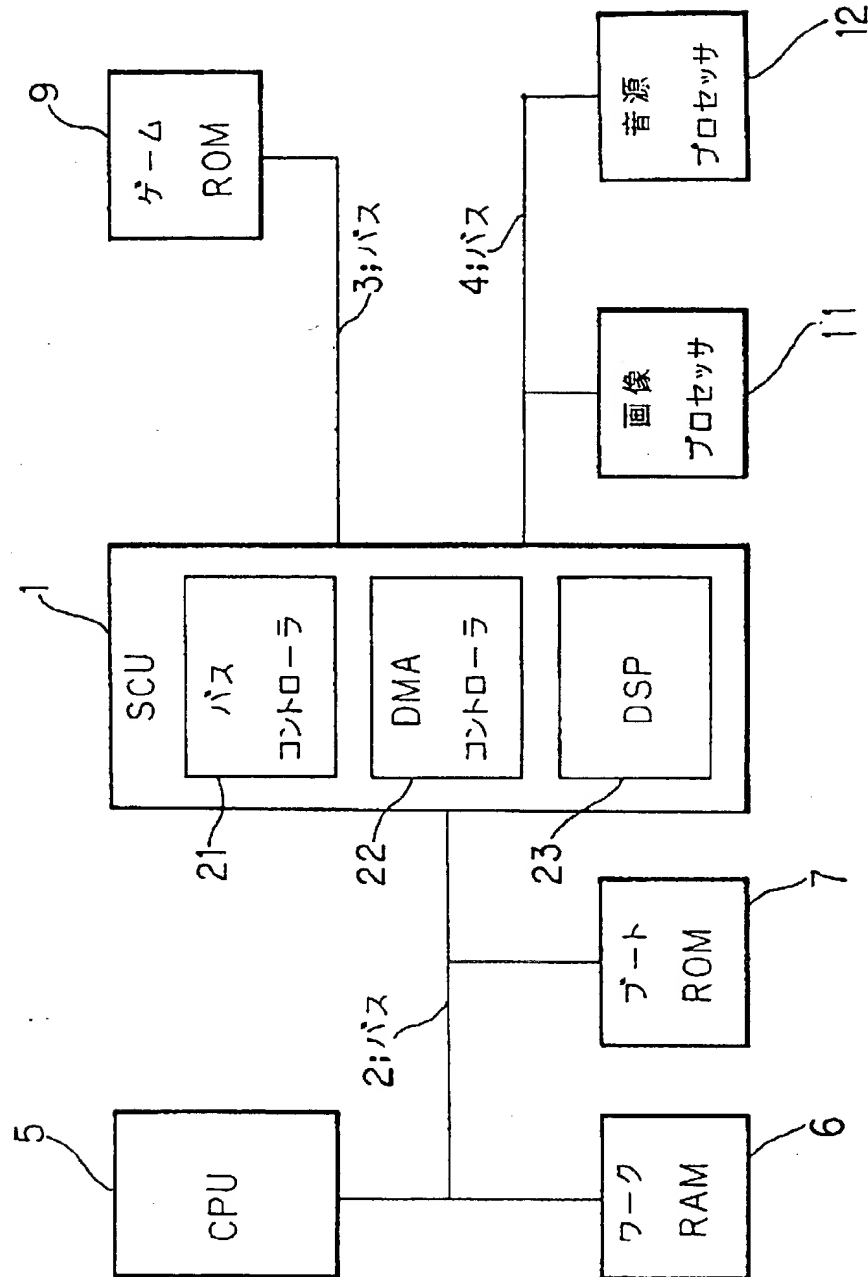
9

10

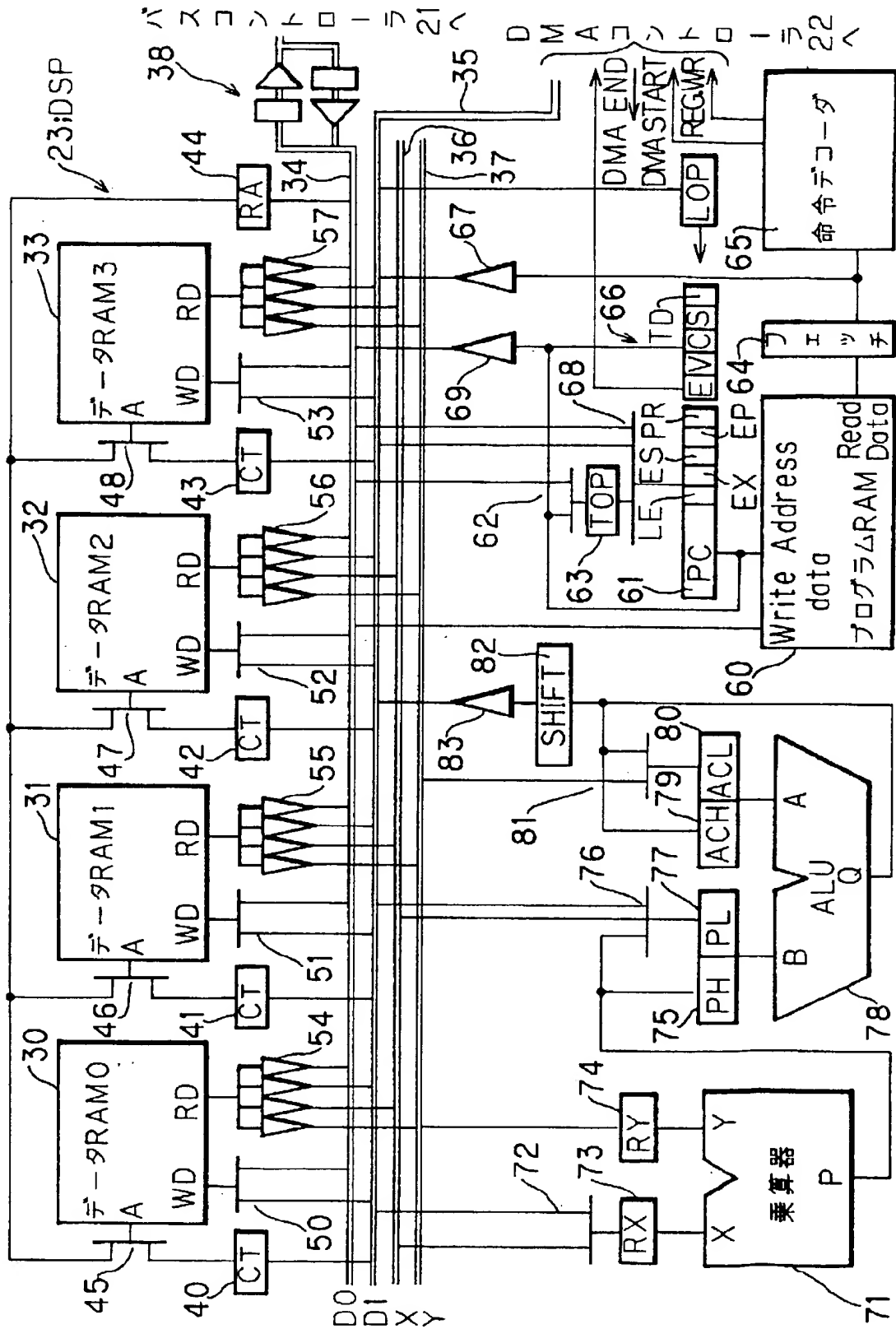
Aコントローラ、23…デジタル・シグナル・プロセッサ、30～33…データRAM、34…D0バス、35…D1バス、36…Xバス、37…Yバス、40～43…アドレスカウンタ、44…アドレスレジスタ、60\*

\*…プログラムRAM、61…プログラムカウンタ、64フェッチ回路、65…デコーダ、71…乗算器、78…ALU

【図1】

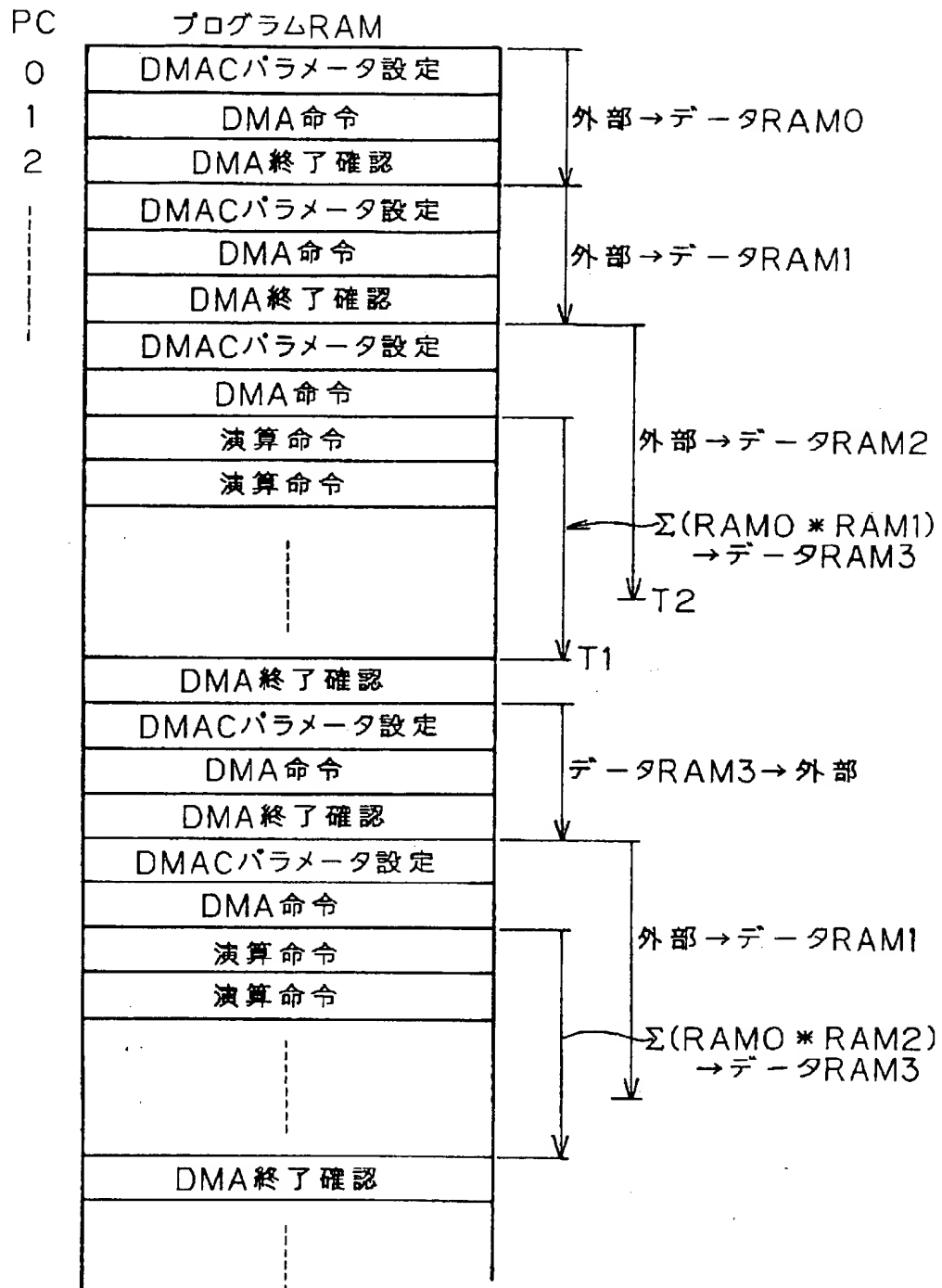


21 < D M A П Л Л О - 22 <

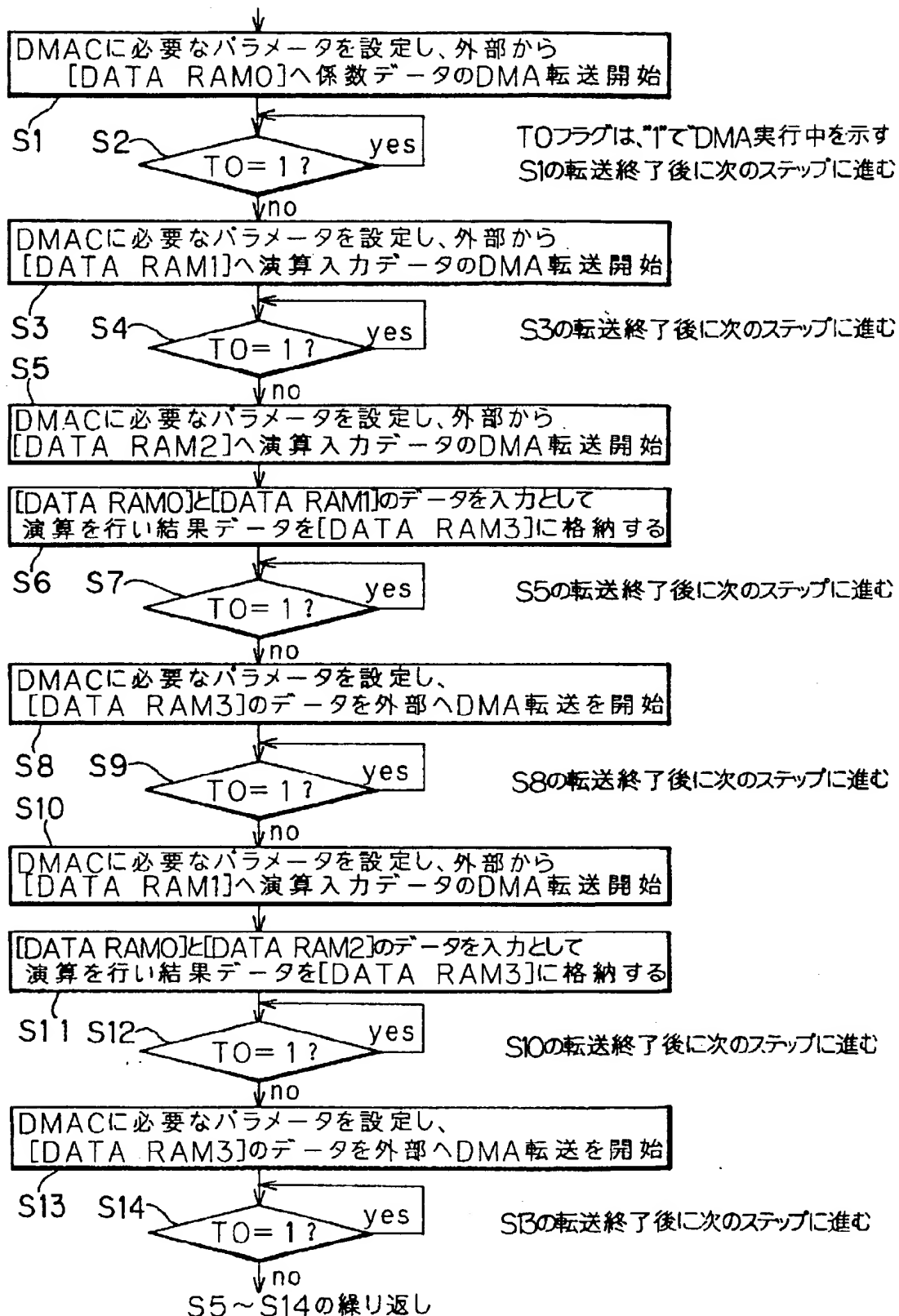




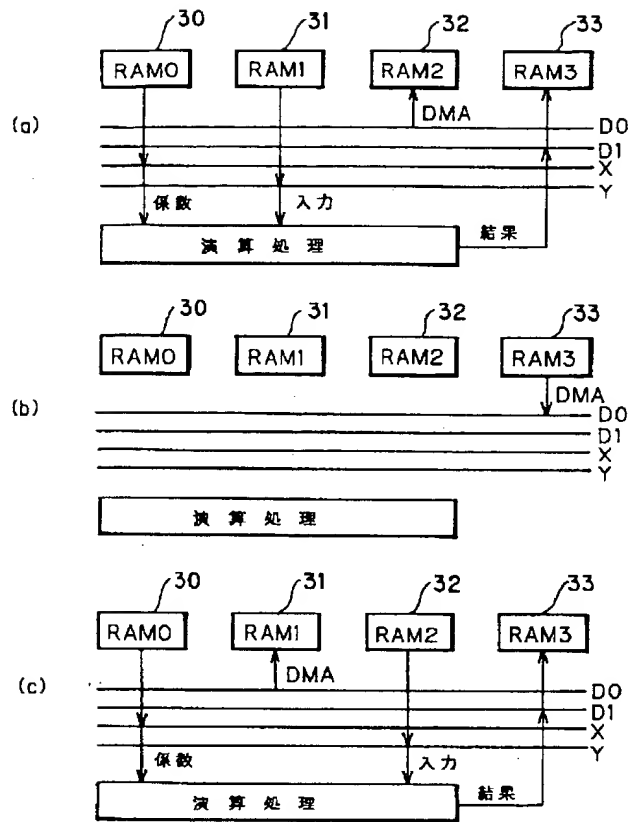
【図3】



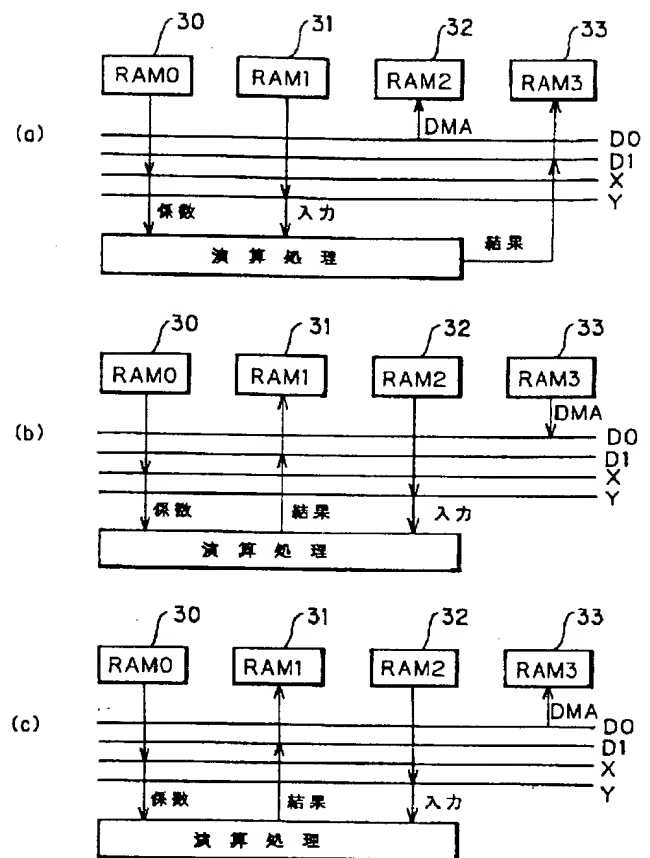
【図 4】



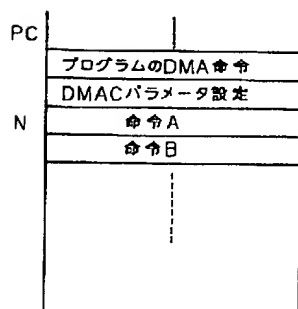
【図 5】



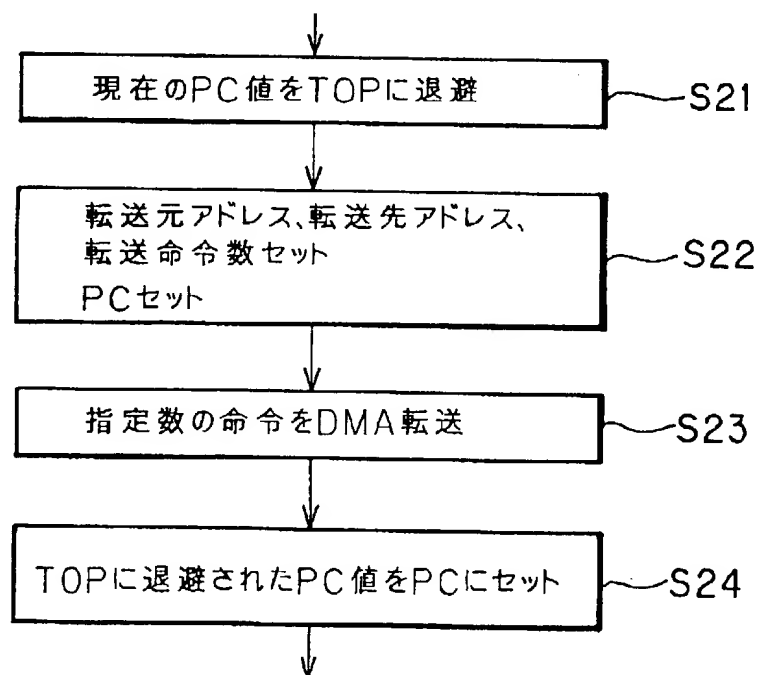
【図 7】



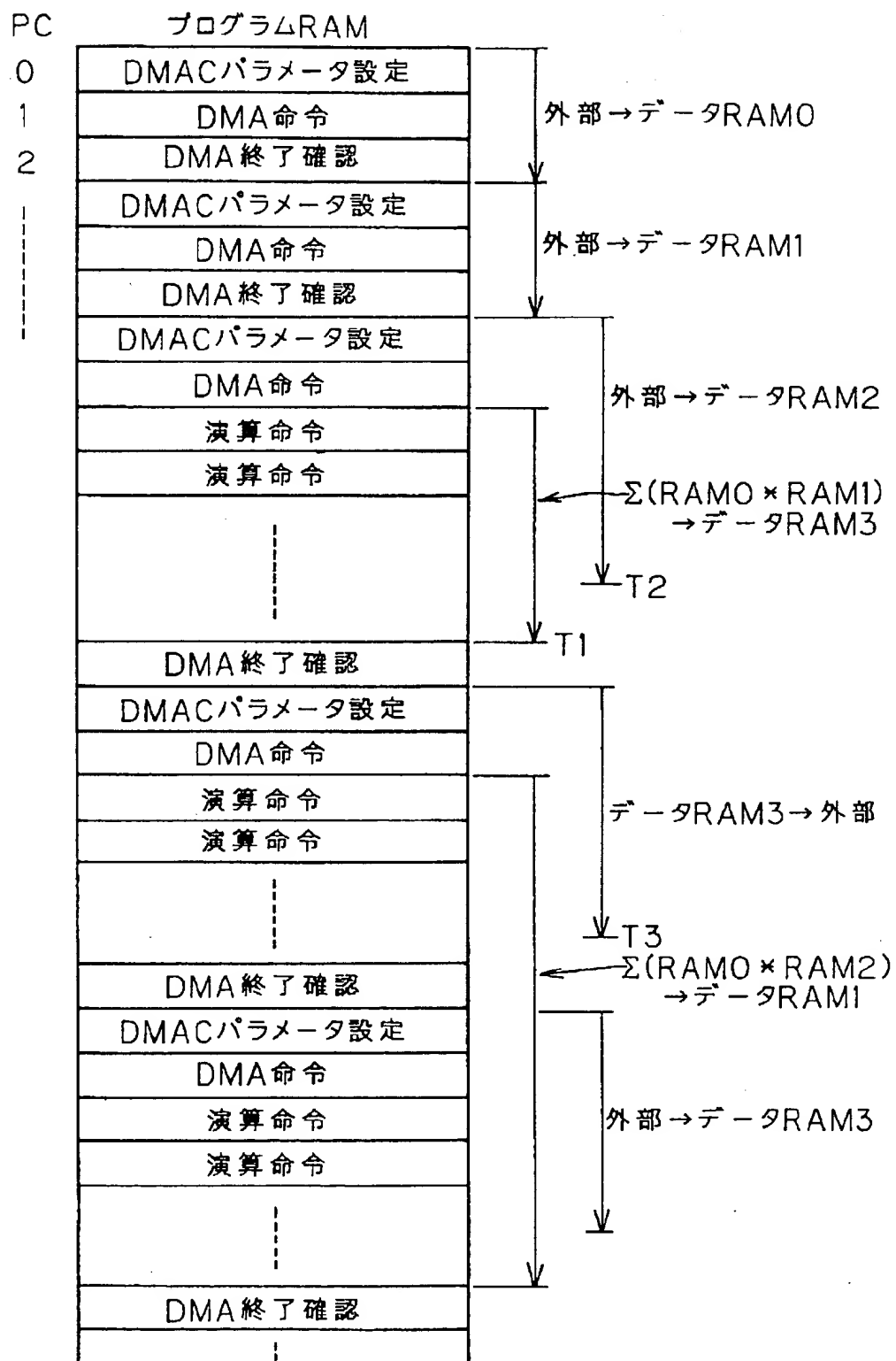
【図 9】



【図 10】



【図6】



【図 8】

